differentiation, performance monitoring, and resource provisioning, to name a few. For example, an operator of an enterprise network may want to prioritize traffic for business critical applications, identify unknown traffic for anomaly detection, or perform workload characterization for designing efficient resource management schemes that satisfy diverse applications performance and resource requirements.

Traffic classification requires the ability to accurately associate network traffic to pre-defined classes of interest. These classes of interest can be classes of applications (e.g. HTTP, FTP, WWW, DNS and P2P), applications (e.g. Skype [310], YouTube [488] and Netflix [331]), or class of service [390]. A class of service, for instance based on QoS, encompasses all applications or classes of applications that have the same QoS requirements. Therefore, it is possible that applications that apparently behave differently, belong to the same class of service [462].

Generally, network traffic classification methodologies can be decomposed into four broad categories that leverage port number, packet payload, host behavior or flow features [31, 244]. The classical approach to traffic classification simply associates Internet Assigned Numbers Authority (IANA) [207] registered port numbers to applications. However, since it is no longer the de facto, nor, does it lend itself to learning due to trivial lookup, it is not in the scope of this survey. Furthermore, relying solely on port numbers has been shown to be ineffective [125, 228, 320], largely due to the use of dynamic port negotiation, tunneling and misuse of port numbers assigned to well-known applications for obfuscating traffic and avoiding firewalls [54, 109, 176, 286]. Nevertheless, various classifiers leverage port numbers in conjunction with other techniques [31, 56, 244, 417] to improve the performance of the traffic classifiers. In the following subsections, we discuss the various traffic classification techniques that leverage ML and summarize them in Tables 4, 5, 6, 7 and 8.

4.1 Payload-based traffic classification

Payload-based traffic classification is an alternate to port-based traffic classification. However, since it searches through the payload for known application signatures, it incurs higher computation and storage costs. Also, it is cumbersome to manually maintain and adapt the signatures to the ever growing number of applications and their dynamics [138]. Furthermore, with the rise in security and privacy concerns, payload is often encrypted and its access is prohibited due to privacy laws. This makes it non-trivial to infer a signature for an application class using payload [54, 138].

Haffner et al. [176] reduce the computational overhead by using only the first few bytes of unidirectional, unencrypted TCP flows as binary feature vectors. For SSH and HTTPS encrypted traffic, they extract features from the unencrypted handshake that negotiate the encryption parameters of the TCP connection. They use NB, AdaBoost and MaxEnt for traffic classification. AdaBoost outperforms NB and MaxEnt, and yields an overall precision of 99% with an error rate within 0.5%.

Table 4 Summary of Payload* and Host Behavior†-based Traffic Classification

Ref.	ML Technique	Dataset	Features	Classes	Evaluation	
					Settings	Results
Haffner et al. [176]*	Supervised NB, AdaBoost, MaxEnt	Proprietary	Discrete byte encoding for first n bytes of unidirectional flow	FTP, SMTP, POP3, IMAP, HTTPS, HTTP, SSH	$n = 64 - 256$ bytes	Overall error rate <0.51%, precision > 99%, recall > 94%
Ma et al. [286]*	Unsupervised HCA	Proprietary: U. Cambridge, UCSD	Discrete byte encoding for first n bytes of unidirectional flow	FTP, SMTP, HTTP, HTTPS, DNS, NTP, NetBIOS, SrvLoc	$n = 64$ bytes, distance metric: PD = 250, MP = 150, CSG = 12%	Error rate: PD $\leq$4.15%, MP $\leq$9.97%, CSG $\leq$6.19%
Finamore et al. [146]*	Supervised SVM	Tstat [439]; NAPA-WINE [268]; Proprietary: ISP network	Statistical characterization of first N bytes of each packet a window of size C, divided into G groups of b consecutive bits	eMule, BitTorrent, RTP, RTCP, DNS, P2P-TV (PPLive, Joost, SopCast, TVAnts), Skype, Background	$C = 80, N = 12, G = 24, b = 4$	Average TP = 99.6%, FP < 1%
Schatzmann et al. [404]†	Supervised SVM	Proprietary: ISP network	Service proximity, activity profiles, session duration, periodicity	Mail, Non-Mail	N/A	Average accuracy = 93.2%, precision = 79.2%
Bermolan et al. [53]†	Supervised SVM	Proprietary: campus network, ISP network	Packet count exchanged between peers in duration ΔT	PPLive, TVAnts, SopCast, Joost	$\Delta T = 5$ s, SVM distance metric $R = 0.5$	Worst-case TPR $\approx$ 95%, FPR < 0.1%

N/A: Not available

Table 5 Summary of supervised flow feature-based traffic classification

Ref.	ML Technique	Dataset	Features	Classes	Evaluation	
					Settings	Results
Roughan et al. [390]	Supervised *k*-NN	Proprietary: univ. networks, streaming service	Packet-level and flow-level features	Telnet, FTP-data, Kazaa, RealMedia Streaming, DNS, HTTPS	$k = 3$, number of QoS classes = 3, 4, 7	Error rate: 5.1% (4), 2.5% (3), 9.4% (7); (#): number of QoS Classes
Moore and Zuev [321]	Supervised NBKE	Proprietary: campus network	Baseline and derivative packet-level features	BULK, WWW, MAIL, SERVICES, DB, P2P, ATTACK, MULTIMEDIA	N/A	Accuracy upto 95%, TPR upto 99%
Jiang et al. [218]	Supervised NBKE	Proprietary: campus network	Baseline and derivative flow-level features	WWW, email, bulk, attack, P2P, multimedia, service, database, interaction, games	N/A	Average accuracy $\approx$ 91%
Park et al. [347]	Supervised REPTree, REPTree-Bagging	NLANR [457]	Packet-level, flow-level and connection-level features	WWW, Telnet, Messenger, FTP, P2P, Multimedia, SMTP, POP, IMAP, DNS, Services	Burst packet threshold = 0.007s	Accuracy $\geq$ 90% (features $\geq$ 7)
Zhang et al. [496]	Supervised BoF-NB	WIDE [474], proprietary: ISP network	Packet-level and flow-level features from unidirectional flows	BT, DNS, FTP, HTTP, IMAP, MSN, POP3, SMTP, SSH, SSL, XMPP	Aggregation rule = *sum*, BoF size	Accuracy 87-94%, F-measure = 80%
Zhang et al. [497]	Supervised RF, Unsupervised *k*-Means (BoF-based, RTC)	KEIO [474], WIDE [474], proprietary: ISP network	Packet-level and flow-level features from unidirectional flows	FTP, HTTP, IMAP, POP3, RAZOR, SSH, SSL, UNKNOWN / ZERO-DAY (BT, DNS, SMTP)	N/A	RTC upto 15% and 10% better in flow and byte accuracy, respectively, than second best F-measure = 0.91 (before update), 0.94 (after update)
Auld et al. [26]	Supervised BNN	Proprietary	Packet-level and flow-level features	ATTACK, BULK, DB, MAIL, P2P, SERVICE, WWW	Number of features = 246, hidden layers = 0-1, 0-30 nodes in the hidden layer, output = 10	Accuracy > 99%, 95% with temporally distant training and testing datasets
Sun et al. [431]	Supervised PNN	Proprietary: campus networks	Packet-level and flow-level features	P2P, WEB, OTHERS	Number of features = 22	Accuracy = 87.99%; P2P: TPR = 91.25%, FPR = 1.36%; WEB: TPR = 98.74%, FPR = 27.7%
Este et al. [140]	Supervised SVM	LBNL [262], CAIDA [451], proprietary: campus network	Packet payload size	HTTP, SMTP, POP3, HTTPS, IMAPS, BitTorrent, FTP, MSN, eDonkey, SSL, SMB, Kazaa, Gnutella, NNTP, DNS, LDAP, SSH	Number of support vectors cf., [140]	TP > 90% for most classes
Jing et al. [223]	Supervised FT-SVM	Proprietary [270, 321]	A subset of 12 from 248 features [321]	BULK, INTERACTIVE, WWW, MAIL, SERVICES, P2P, ATTACK, GAME, MULTIMEDIA, OTHER	SVM parameters automatically chosen	Accuracy up to 96%, error ratio $\downarrow$ 2.35 times, avg. computation cost $\downarrow$ 7.65 times
Wang et al. [464]	Supervised multi-class SVM, unbalanced binary SVM	Proprietary: univ. network	Flow-level and connection-level features	BitTorrent, eDonkey, Kazaa, pplive	N/A	Accuracy 75-99%

N/A: Not available

Table 6 Summary of unsupervised flow feature-based traffic classification

Ref.	ML Technique	Dataset	Features	Classes	Evaluation	
					Settings	Results
Liu et al. [283]	Unsupervised *k*-Means	Proprietary: campus network	Packet-level and flow-level features	WWW, MAIL, P2P, FTP (CONTROL, PASV, DATA), ATTACK, DATABASE, SERVICES, INTERACTIVE, MULTIMEDIA, GAMES	$k = 80$	Average accuracy $\approx$ 90%, minimum recall = 70%
Zander et al. [492]	Unsupervised AutoClass	NLANR [457]	Packet-level and flow-level features	AOL Messenger, Napster, Half-Life, FTP, Telnet, SMTP, DNS, HTTP	Intra-class homogeneity (H)	Mean accuracy = 86.5%
Erman et al. [136]	Unsupervised AutoClass	Univ. Auckland [457]	Packet-level and flow-level features	HTTP, SMTP, DNS, SOCKS, IRC, FTP (control, data), POP3, LIMEWIRE, FTP	N/A	Accuracy = 91.2%
Erman et al. [135]	Unsupervised DBSCAN	Univ. Auckland [457], proprietary: Univ. Calgary	Packet-level and flow-level features	HTTP, P2P, SMTP, IMAP, POP3, MSSQL, OTHER	$eps = 0.03$, $minPts = 3$, number of clusters = 190	Overall accuracy = 75.6%, average precision > 95% (7/9 classes)
Erman et al. [138]	Unsupervised *k*-Means	Proprietary: univ. network	Packet-level and flow-level features from unidirectional flows	Web, EMAIL, DB, P2P, OTHER, CHAT, FTP, STREAMING	$k = 400$	Server-to-client: Avg. flow accuracy = 95%, Avg. byte accuracy = 79%; Web: precision = 97%, recall = 97%; P2P: precision = 82%, recall = 77%

N/A: Not available

Their ML models are scalable and robust due to the use of partial payloads, and unidirectional flows and diverse usage patterns, respectively. The unidirectional flows circumvent the challenges due to asymmetric routing. In comparison to campus or enterprise networks, residential network data offer an increased diversity, with respect to, social group, age and interest with less spatial and temporal correlation in usage patterns. Unfortunately, performance of AdaBoost traffic classifier deteriorates with noisy data [176] and their approach requires a priori knowledge about the protocols in the application classes.

Ma et al. [286] show that payload-based traffic classification can be performed without any a priori knowledge of the application classes using unsupervised clustering. They train their classifiers based on the label of a single instance of a protocol and a list of partially correlated protocols, where a protocol is modeled as a distribution of sessions. Each session is a pair of unidirectional flow distributions, one from the source to the destination and another from the destination to the source. For tractability, the sessions are assumed to be finite and a protocol model is derived as a distribution on n byte flows, rather than pair of flows.

In product distribution (PD) protocol model, the n byte flow distribution is statistically represented as a product of n independent byte distributions, each describing the distribution of bytes at a particular offset in the flow. Similarly, in the Markov process (MP) protocol model, nodes are labeled with unique byte values and the edges are weighted with a transition probability, such that the sum of all egress transition probabilities from a node is one. A random walk through the directed graph identify discriminator strings that are not tied to a fixed offset. In contrast, the common substring graphs (CSG) capture structural information about the flows using longest common subsequence. A subsequence in a series of common substrings that capture commonalities including the fixed offsets in statistical protocol modeling.

Finally, the authors perform agglomerative (bottom-up) hierarchical clustering analysis (HCA) to group the observed protocols and distinguish between the classes of interest. They employ weighted relative entropy for PD and MP, and approximate graph similarity for CSG, as the distance metric. In evaluation, the PD-based protocol models resulted in the lowest total misclassification error, under 5%. Thus, there is a high invariance at fixed